Chroma

Newsletter of the Australian Computer Music Association, Inc.
PO Box 4136 Melbourne University VIC 3052

Number 5
June 90
ISSN 1034-8271

In This Issue

ACMA CD - Final Call for Pieces

As you know, ACMA is producing a CD of recent electro-acoustic works by members later this year. If you are interested in including one of YOUR works, get in touch as soon as possible, as there is still space available on the CD. The Australian Centre for Art and Technology (ACAT), directed by David Worrall, generously offered to fund the project, but the committee decided it would be best at this stage to "live free or die", i.e. remain an independent body. As ACAT intends going ahead with their great CD program anyway, we'll hopefully see more Australian electro-acoustic music available this year than ever.

HMSL 4.0 /Frog Peak Music

Email last week from Larry Polansky announced the release of version 4 of HMSL. No further details yet, but as users will know, MACH 2 Forth for the Mac has been dumped and a whole new version of Forth has been written, specifically for use with HMSL. Frog Peak Music now lists Chroma in its catalogue and we are hoping to be able to return the favour and distribute Frog Peak stuff here in Australia (including HMSL).

In this issue we have the first of an excellent series on Copyright issues and computer music in Australia by Shireen Tippett; a follow up by Jim Sosnin to his article on programming the Amiga Sound Chip using Basic, with an extended example; and a review of the electro-acoustic music disks from the Anthology of Australian Music on Disk, whose long awaited release was announced in the last issue.

TISEA

The **Third International Syposium on Electronic Art** will be held in Australia in 1992, and ACMA, along with the Australian Centre for Art and Technology (ACAT) and the Australian Network for Art and Technology (ANAT) has been asked to participate in putting it together.
This is an exciting opportunity for all of the Electronic Arts in Australia - to meet with practitioners from all over the world and display and discuss our work. More in the next issue, but if you have any ideas in the mean time, get in touch.
The Second International Syposium on Electronic Art is to be held in Gröningen, the Netherlands from November 12-17 1990. Topics include computer graphics and animation, electronic and computer music, computer poetry, computer assisted dance, robotics, aesthetics, encoding standards for electronic music, synchronization of images and sound, real time systems and more. As well as conference papers, there will be exhibitions, workshops, performances/concerts, film and video shows as well as events open to the general public.

Contributions - We need them! If you have an idea for an article, review, interview, news item or anything you think Chroma should or could carry, let's hear it – give Graeme Gerrard a ring on (03) 344 4127 (W) or (059) 62 5899.

Membership - It's time to renew your membership for 1990, if you haven't done so already. There's a membership renewal form enclosed so send it off with your $10 today. If you have already renewed, pass the application on to someone else you think will be interested. We need to increase our membership to achieve "critical mass" - enough people to really make ACMA worthwhile. If you have any ideas along this line please let us know.

Dear Editor,

MIDI file format

Just wanted to know if you or any of the Chroma readers could tell me where I could get a description of the standard MIDI file format?

Robert Zurinski
ph. (02) 76 6400

Dear Robert,
The International MIDI Association sells the MIDI file format specification, either with the MIDI specification itself, or separately. I don't remember details of costs, but you can find out from:

I.M.A.
5316 W. 57th Street
Los Angeles, CA 90056
ph. (213) 649 - 6434

Something that may interest you in the mean time is TabConverter, by David Zicearelli, one of the authors of M. TabConverter converts between a textual representation of MIDI events and format 0 MIDI files. Each parameter of the MIDI event (e.g. MIDI channel no., start time, duration, pitch and velocity, in the case of Note ON events) is separated by a tab character, and is quite readable and editable with standard word processors. (Word, MacWrite, etc. on the Macintosh). Moreover, TabConverter is FREE, and can be freely copied, but not sold. There may be a version for IBM as well.

From Singapore

I have received a rather long letter from Joseph Peters, Assistant Director of the Centre for Musical Activities at the National University of Singapore, which I can only paraphrase here. Last year Joseph and his students staged Synthecom, an extravagant multi-media presentation of computer music, dance, and MIDI controlled light show, using a locally designed MIDI-to-light interface. Using Macintosh SEs running Performer, the sequences in the score contained 24 tracks of light control. He speaks enthusiastically of Chroma and the standard of members' contributions. If anyone would like to get in touch with him, his address is:

Joseph Peters
Centre for Musical Activities
National University of Singapore
Kent Ridge
Singapore 0511

Chroma is edited by Graeme Gerrard. © 1990 Australian Computer Music Association, Inc. and the authors.

Automatic Composition

James Ward writes: [*again I am paraphrasing a longer letter*, Ed.]

. . . I am working on a system designed to compose musical pieces fully automatically. It works in three stages, the first is a composing algorithm which works on the principle of constrained random choises. The output of this stage is a high level description of a musical piece, which is then interpreted by stage two to produce a text file "score". The last step in the process is the realisation of the score by my 'score player'. This is a Turbo Pascal program which plays text files using the Macintosh square wave synthesiser and Sound Driver. The source code, documentation and examples is available for $10 or a disk and $5. It runs as a stan alone application, so you don't need Turbo Pascal to run it. I would like to extend this program to use the Macintosh Sound Manager or generate MIDI files directly.

Enquiries to:
James Ward
8 Karoola Cres.
Caringbah NSW 2229
ph (02) 524-9711

Music4C for Macintosh

Music4C is a C variant of MUSIC4BF, ancient sound synthesis program written by Hubert S. Howe Jr. and Godfrey Winham at Princeton Uni in the mid 1960s. It runs on Macintosh SE/30 and II series computers (i.e. with 68881/2 floating point processors) and exists as a THINK Lightspeed C project (v4.0) .

I have worked on this off and on for almost 18 months and since announcing it in the Computer Music Journal a couple of issues ago, have distributed about 30 copies around the world - from IRCAM to Berkelee, Portugal to Switzerland, New York to Perth.

Music4C has software modules such as filters, oscillators, envelop controllers, waveshapers, formant generators, etc. that you use to build a signal generator or signal processor. While being rather slow for synthesis of complete compositions, though it has been done, it is particularly useful for soundfile manipulation and synth algorithm testing.

Music4C is free, but I ask $10 for diskette, postage and handling. It comes with sources, examples, documentation and a list of other users. Its sound file formats include Audio Interchange File Format (AIFF), Sound Designer (I and II), and standard interleaved floating point or integer, and you can use it with Alchemy and Sound Designer.

Please contact Graeme Gerrard (03) 344 4127.

The Relevance of Copyright to Synthesis, Sampling, and Computer Generated and Assisted Compositions (Part I)

Shireen Tippett

Introduction

"Technological change has fundamentally altered the context and assumptions of many of our laws and institutions. Not least among these, it has altered the relationship between creators and the users of their works. Nevertheless, the right of creators to control the use of their works, and the right of users to fair access will always be a fundamental tension in copyright protection." (Govt. of Canada 1984, author not specified).

My intention here is to concentrate on three specific areas of technology – synthesis, sampling and computer-generated compositions, highlighting the ways in which they are covered either specifically or indirectly by Australian Copyright Law. In order to do this, it will be necessary firstly to look at the Australian Copyright Act (1968) and its recent amendment (1989), and briefly outline the fundamental sections which prove to be relevant to these three distinct aspects of technology. In evaluating the efficiency of this Act, it will also be necessary to draw upon overseas examples of Copyright protection.

To date, copyright problems in relation to Synthesis have received little attention, but it has become quite clear that certain specific areas require investigation. These areas consist mainly of patch copyright, shareware, pirate sounds, legal precedents, the ethics of synthesis, and possible solutions for the present, and projected future developments.

With the advent of 'Rap Music', the issue of Sampling and Copyright has been receiving considerable attention from musicians, lawyers and the media. The major areas that are prevalent are the use of prerecorded material for samples, pirating of samples on the market, possible performing rights infringement with the unauthorised sampling of live performances, various ethical questions stemming from the above areas, proferred solutions, and problems expected from future developments.

The final technological development to be considered involves the ownership of computer-generated works; the use of copyrighted material to generate new works with the aid of the computer; the use of copyrighted programs to generate new programs and in turn, new works; computer sound synthesis and sound processing; reproduction of a copyrighted work with the aid of a computer; ethical questions raised by this rapidly developing area and an outline of possible solutions.

Copyright Law

There are seven main areas in Copyright Law that are relevant to the areas of Synthesis, Sampling and Computer-generated compositions. These are:-

1) Definition of Copyright;
2) Ownership of Copyright;
3) Copyright in Original Works;
4) Copyright in Sound Recordings;
5) Performing Rights;
6) Fair Use and
7) Infringement of Copyright Works.

Copyright Defined

Copyright (Halloran, 1979, 54) is defined as "a property right comprised of[sic] a set of legally enforceable privileges granted by law to creators of artistic works to encourage them to create". Ploman & Hamilton(1980) provide six reasons for the existence of copyright :-

1) common social justice : the author should be able to benefit from the fruits of his labour with royalties being equivalent to the worker's salary.
2) cultural progress : to encourage the author to create new works and therefore enrich the country's store of music etc.
3) economic grounds : investment necessary for the creation of works (costs involved).
4) moral grounds : personal expression of thought (which is therefore the intellectual property of the author).
5) national prestige : cultural heritage of a country.
6) export, import and cultural interchange.

Ownership of Copyright

1) Original Works

Section 35(2) of the Copyright Act(1968) states that "the author of literary, dramatic, musical or artistic work is the owner of any copyright subsisting in the work...". One exception given in Section 35(6) is where a musical work "is made by the author in pursuance of the terms of his employment by another person under a contract of service or apprenticeship", then "that person is the owner of any copyright subsisting in the work...".

2) Sound Recordings

The maker of the sound recording is the owner of any copyright subsisting in that recording where -

" (a) a person makes, for valuable consideration, an agreement with another person for the making of a sound recording by the other person; and

(b) the recording is made in pursuance of the agreement, the first-mentioned person is, in the absence of any agreement to the contrary, the owner of any copyright subsisting in the recording..." (Copyright Act, 1968, s.97)

Copyright in Original Works

Section 31 of the Australian Copyright Act(1968) states that "For the purposes of this Act...copyright, in relation to a work, is the exclusive right –

a) in the case of a literary, dramatic or musical work, to do all or any of the following acts:
   i) to reproduce the work in a material form;
   ii) to publish the work;
   iii) to perform the work in public;
   iv) to broadcast the work;
   v) to cause the work to be transmitted to subscribers to a diffusion service;
   vi) to make an adaptation of the work;
   vii) to do, in relation to a work that is an adaptation of the first-mentioned work, any of the acts

specified in relation to the first-mentioned work in sub-paragraphs i) to v), inclusive."

Section 32 states that copyright subsists in an original work that is unpublished and of which the author –

" a) was a qualified person at the time when the work was made or

b) if the making of the work extended over a period - was a qualified person for a substantial part of that period." (Copyright Act, 1968). Qualified here means an Australian citizen, an Australian protected person or a person resident in Australia. In a work that is published –

" a) copyright subsists in the work; or

b) if copyright in the work subsisted immediately before its first publication - copyright continues to subsist in the work, if, but only if-

c) the first publication of the work took place in Australia;

d) the author of the work was a qualified person at the time when the work was first published; or

e) the author died before that time but was a qualified person immediately before his death.(Copyright Act, 1968,s.32)

In the US Copyright Act, there are two main prerequisites set by law for obtaining copyright ownership in a composition. Firstly, the song must be original to the author, which means that the author created the work. Secondly, it must be fixed in a tangible medium of expression, which means that the work must be on tape, paper or other means whereby it may be perceived for longer than a very short period of time (Halloran, 1979). These prerequisites raise the question of what cannot be copyrighted. Songs that are not 'fixed in a tangible medium of expression' (for example, if they are fixed only in the composer's memory), and song titles are not copyright protected. Ideas also don't afford protection by copyright, although the expression of these ideas form the basis for copyright.

Copyright in Sound Recordings

Section 85 of the Copyright Act(1968) states that copyright in relation to a sound recording, "is the exclusive right to do all or any of the following acts:

a) to make a copy of the sound recording;
b) to cause the recording to be heard in public;
c) to broadcast the recording."

Sound recordings in which copyright subsists :

" 1) Subject to this Act, copyright subsists in a sound recording of which the maker was a qualified person at the time when the recording was made.

2) Without prejudice to the last preceding sub-section, copyright subsists, subject to this Act, in a sound recording if the recording was made in Australia.

3) Without prejudice to the last two preceding subsections, copyright subsists, subject to this Act, in a published sound recording if the first publication of the recording took place in Australia." (Copyright Act, 1968, s.89)

Performing Rights

Until the amendment of 1989, the Australian Copyright Act had failed to address the issue of performers' rights, despite the fact that the UK had a Statute in Force (Dramatic and Musical Performers' Protection Act) since 1958. This section is found in Part XIa of the 1989 Copyright Amendment Act. In effect, what this section means for musical performers' is that no unauthorised recordings may be made of their performances, with certain exceptions (see Appendix A). Understandably, such an amendment has serious connotations in relation to sampling live sounds.

Fair Use

Fair Use provisions are not as clear in the Australian Copyright Act as they are in Overseas models such as the United States Act. In the Australian Copyright Act, reference is made firstly to 'Fair Dealing' with regards to purposes of research or study, criticism or review, reporting news, judicial proceedings or professional advice, backup copy of a computer program and collections for use by places of education; and secondly to 'Reasonable Portion', which applies purely to a published edition of a literary, dramatic or musical work (score) that is greater than ten pages (Copyright Act, 1968, s.10(2)). There is also a clause in Section 14 (Copyright Act, 1968) regarding "Acts done in relation to substantial part of work or other subject-matter deemed to be done in relation to the whole

(a) a reference to the doing of an act in relation to a work or other subject-matter shall be read as including a reference to the doing of that act in relation to a substantial part of the work or other subject-matter; and

(b) a reference to a reproduction, adaptation or copy of a work shall be read as including a reference to a reproduction, adaptation or copy of a substantial part of the work, as the case may be."

In other words, an infringement may occur regardless of the amount used from the copyright protected work. 'Substantial part' is not defined anywhere in the Act, so it would appear that any unauthorised use of copyright material may be considered to be a breach of copyright. This is quite different to the US Copyright Act which outlines the permissable use of a small part of a copyright work, the amount being determined by certain factors :-

1) the purpose and character of the use (including whether the use is of a commercial nature or for non-profit educational purposes);
2) the nature of the copyrighted work itself;
3) the amount and substantiality of the portion of the copyrighted work used; and
4) the effect of the use on the commercial value of the copyrighted work.

With regard to the Australian Act, although there don't appear to be fair use provisions as such, it is still necessary in arguing a breach of copyright to prove that an infringement has in fact taken place. Some use may have been made of copyrighted material, but if this is not clearly evident, it will be extremely difficult to prove an infringement took place.

Infringement of Copyright Works

As proven in the 1963 case of Francis Day and Hunter Ltd. versus Bron outlined by Laddie, Prescott and

Vitoria (1980) and involving United States Copyright, "For there to be an infringement of copyright, there must be present two elements : first, there must be sufficient objective similarity between the infringing work and the copyright work, or a substantial part thereof, for the former to be properly described, not necessarily as identical with, but as a reproduction or adaptation of the latter; secondly, the copyright work must be the source from which the infringing work is derived."

Also, in order to prove an infringement in the United States, the copyright owner or his/her representative must show that the infringer did in fact copy the protected work; that the infringer did in fact have access to the work and that the infringer made a substantial taking from the work which need not have been a conscious act. (Gunnar, Hearn & Halloran, 1983) The copyright owner must also prove that they in fact own the rights.

This is similar to the situation in Australia, with the exception that the copyright owner is not entitled to any damages, only costs if the infringer "was not aware, and had no reasonable grounds for suspecting, that copyright subsisted in the work or other subject-matter to which the action relates." (Copyright Act, 1968, s.116(2))

Part II in Chroma 6 will cover ***COPYRIGHT*** *in relation to* ***SYNTHESIS AND SAMPLING.***
Part III in Chroma 7 covers ***COPYRIGHT*** *and* ***COMPUTER-GENERATED*** *and* ***ASSISTED Compositions.***

Equipment for Sale or Swap

Tascam 224 4 track reel-to-reel recorder
Yamaha CX5 music computer
Fender Jazz Bass
Roland Cube Bass Amp 100watt, 15" speaker
Casio MT30 digital keyboard
Dr Rhythm Drum Machine

Will swap for:
Yamaha Tx802 FM synth module
Stereo Power amp (100watt or greater)
Guitar amplifier

contact David Hirst (03) 479 1502 or 481 5406

Electro-Acoustic Music on the Anthology of Australian Music on Disc.

Reviewed by Warren Burt

CSM:4— Cary, Gerrard, Parish, Worrall, Pompili, Riddell.
CSM:5—Worrall, Tahourdin, Exton, Fredericks, Cary.
CSM:6—Burt, Milsom, Althoff, Mann, Mumme, Chesworth.
CSM:13—de Haan/Schiemer, Wesley-Smith, de Haan, in addition to instrumental works by Sitsky and Hair.
Available for $22 each from:

School of Music
Canberra Institute of the Arts
GPO Box 804, Canberra, ACT 2601;

or from

Sounds Australian,
PO Box 49, Broadway, NSW 2007.

The long-delayed, controversy-racked, and problem plagued Anthology of Australian Music on Disc is finally available. Fifteen compact discs cover a wide spectrum of Australian music, and although there are some glaring omissions from the set, none of the organizers of the project makes any claims that this release tells the complete story. A further 2 or 3 releases per year are currently being planned, and if the project can be sustained at that level for the next decade, a project that is already extremely valuable will make itself indispensable.

Of the fifteen discs, three are devoted exclusively to electro-acoustic works, while a fourth, featuring trombonist Simone de Haan, has 3 electro-acoustic works. Due to the length of time the set took to issue, none of the works, save one, is less than 5 years old. (The good folks at CSM now realize that a 15 disc set is not something a few dedicated and overworked people can do in their off-hours!) As a portrait of the state of electro-acoustic music in Australian the early to mid-80s, however, the set is invaluable. To the best of my knowledge, only two of the works have been available before (David Worrall's *...with fish scales scattered...* and Brian Parish's *Contours, Clowns, and Shadows*, both on the NMA label), and the spectrum covered by the works is wide, from academic timbral research to simple tunes, from concrete to analog to digital, from socially engaged work to work that is highly formalized. In addition to the works from 1979-85, the set also includes two earlier pieces, Tristram Cary's *Continuum* (1969) and John Exton's *Breathing Space* (1972), both of which are classics in their genres. All of the discs sound wonderful, all of them are sequenced very well, and each of them, as a disc on its own, makes for really pleasurable listening. Congratulations are in order to David Worrall and Niven Stines for their immaculate technical work.

From my perspective, the three electro-acoustic discs break down roughly into three categories – the formalist and mathematical; the mystical; and the experimental and socially engaged. CSM:4 is the formalist disc, starting off with Tristram Cary's engaging, totally serialized, microtonal *Nonet*, made in 1979 at Stanford, and continues with Graeme Gerrard's *Strings of Token Strings* (1984, La Trobe), where he works with his compositional language, COMPOST. Next comes Brian Parish's *Contours, Clowns and Shadows* (1982, La Trobe) which came out of his work with graphic score inputs for computer synthesis. Both David Worrall's *...with fish scales scattered...* (1982, Melbourne Uni.) and Claudio Pompili's *Medieval Purity in a Bed of Thorns* (1981, Adelaide Uni.) deal with ideas of permutation, stasis, and the golden mean, while Alistair Riddell's *Atlantic Fears* (1983), uses his computer controlled piano to explore permutations developed by composing routines. Although all the works are primarily concerned with structure, all of them make for nicely sensual listening.

Three of the five works on CSM:5 have overtly mystical or philosophical concerns, while a fourth, David Worrall's *Butterflies Flutter By* (1987, Canberra School of Music) also nods in that direction. The only exception to this theme is Peter Tahourdin's *San Diego Canons* (1983), a likeable percussion work made by processing and remixing tapes made in San Diego in 1980. John Exton's *Breathing Space*, (1972, Cardiff) mixes human sounds, mechanical sounds and electronic sounds into an effecting noise-inflected soundscape. Some of the human sounds – gasps and sighs – that emerge from the texture had me sitting up to take notice, so effective was the illusion of a person at that point in the sonic space. Ian Frederick's *Some Quiet Graveyard* (1984, Sydney Uni) uses his homemade controllers to provide timbral inflection and spatial positioning to sounds that would not otherwise be possible. The timbral inflections particularly give the sounds an unusually expressive quality. Inflection, not normally a part of "electronic music," is here used as one of the main elements of the music. Tristram Cary's *Continuum* (1969), made in his then state-of-the-art home studio, is a classic analog piece with no beginning and no end, but an intricate structure in between. Like the Exton and the Fredericks, it shows the composer dealing with ideas of cosmic structure, trying to express his dealing with those ideas in a musical manner.

CSM:6 starts with my *An Eminently Performable Piece* (1982), made with the Serge analog synthesizer, which explores the emotional affects of a series of ancient Greek modes, and continues with Peter Milsom's *Reflective Transience* (1983, Adelaide Uni.) which features some of the most attractively gritty, rasping timbres in the entire set. Ernie Althoff's *51 Flexibles for Gramophone Users* (1985, Rainer Linz studio) is a happy victim of technological obsolescence. The piece consists of Ernie "scratching" phonograph records while he reads a text giving the listeners various options for ways they can play with their phonographs to alter what they hear. Conceived specifically for the medium of black vinyl, the change of the set to CD has given the piece a new dimension: as an epitaph to the medium it was conceived for. In Chris Mann's *snodger (the mirror)* (1981, La Trobe), Chris reads a text of his both in and out of phase with a tape of his voice reading the same text computer processed so that each word is filtered and mixed with a backwards version of itself. The end result is a series of intricately shaped utterances, some of which make sense, some of which don't, but all of which are very pretty. Peter Mumme's *Metafork* (1985) and David Chesworth's *Tissues for Issues* (1984), were both made in their home studios, and both accompany simple tunes with other materials. In Peter's case, the accompaniment is bird song, in David's case, a cut up of voices from radio. Originally conceived for broadcast, the effect of David's selection of voices and issues would have been much different if heard, unawares, on a radio. Six years later on a CD, it emerges as a more politically engaged, and a much more humorous work than it might have at the time.

CSM:13 is a disc featuring the remarkable trombonist Simone de Haan, both solo and in collaboration with others. Of the five works on the album, the works by Larry Sitsky and Graham Hair are beyond the scope of this review (but both are well worth listening to!), while three works show Simone's continuing engagement with electro-acoustics. First is an *Improvisation* (1983) made in Llewelyn Hall, Canberra, with Simone on trombone and Greg Schiemer on his wonderful tupperware gamelan. Unfortunately, this is the only tape that was ever made of the tupperware gamelan, and though the piece is fine on its own, it doesn't show nearly the full capabilities of that remarkable instrument. Is there anyone out there specializing in early music on authentic instruments who would like to reconstruct the tupperware gamelan for posterity? Martin Wesley-Smith's *Pat-a-Cake* (1980, Sydney Con.) continues his obsessions with nursery rhymes, Rev. Dodgson, and the CMI, to make a work for trombone and sampled trombones, while Simone's own Major C (1980, Queensland Con.) explores the lush timbral resources available by multi-layering trombone sounds on tape.

I have only one complaint with the set, in fact, but that is a big one. The set was originally designed to be sold as a unit, costing $300, with all the program notes contained in a separate booklet, available on it's own for $30. Unfortunately, with the separate CDs, the only information available is titles and timings. There are no notes whatever accompanying any of the discs. As a composer of one of the works in the set, I had to travel out to the Monash Uni. library to read the notes on my (and everybody else's) piece in order to write this review! I understand that this lack of notes with the individual CDs was due to lack of finances and staff time, but not having notes does smack of that obnoxious and obsolete idea that music is a "pure art" which exists "above" any other than purely sonic considerations. This lack is especially keenly felt in the case of the electro-acoustic works, each of which presents both an aesthetic and a technical argument. Each of these works has a world of concerns which are both beyond their mere sounding surfaces and which also, paradoxically, make the works what they are. Without notes, we are denied half the music – the ideas half.

Still, the set is a brave start at documenting an area largely ignored until now, so we can only be grateful for the hard work that has gone into it, and hope perhaps, that in the future, some sort of inserts with notes can be arranged for the individual CDs.

AmigaBasic Real-Time Sound Synthesis (continued)

Jim Sosnin

Here are some more ideas, and another example program, following the AmigaBasic Mouse Theremin in the last issue. First some news: soon after the mail out of the last issue, in which I lamented the Amiga audio hardware being an underused resource, Warren Burt gave me a copy of the shareware program RGS (Real-time Graphical Synth), written in the JForth language by J Henry Lowengard in New York. Briefly, RGS allows the user to mouse-draw a sonogram on the Amiga screen, then to hear the corresponding sounds, either as the drawing takes place, or with better fidelity after a short delay. You can think of the sonogram as a graphic score with time on the X axis and pitch on the Y axis; sound events are represented by connected or unconnected groupings of three basic forms: dots/lines, blobs and ramps, all different in their temporal and timbral control.

RGS has other features, including MIDI output, that make it a useful compositional tool, or even a performance instrument. Every musician with an Amiga really should have a look at this program; if you would like a copy, please contact me, either directly, or c/o Chroma. Then someone can write RGS the review it deserves!

Now back to Basic, which I am using to explore the characteristics of the Amiga DMA hardware. My second offering (see Program Example 2) demonstrates the modulation capabilities of the oscillators, both AM and FM. When you run the program, you can select AM, FM, or both, and then play notes with the mouse button, moving the mouse to control duration, frequency or volume.

In AM mode, mouse X (horizontal) controls the decay duration of an amplitude envelope which starts at maximum amplitude and ramps down linearly to zero; mouse Y (vertical) controls the pitch, completely independent of the duration (this would not be possible if, instead of using modulation, we were scanning a lookup table of an already enveloped waveform). Pressing the button plays one note only, with the program in its original form, but a simple change described later allows the envelope to keep repeating as long as the button is held down; this can be useful for generating interesting sound combinations in the 'Both' mode, since the FM mode continues sounding in this way.

In FM mode, mouse X controls the period of the modulator; it is best to think of this as duration when the mouse is towards the right, since the period is long enough for perception of carrier pitch sweeps, and to think in terms of frequency when the mouse is towards the left, when the period is short enough to change the timbre of the carrier. Mouse Y controls the volume of the carrier; it would have been nice to have mouse Y control the carrier centre frequency, but unfortunately it is not possible to have program control of an oscillator frequency if another oscillator is modulating its frequency already. This is one of the limitations of Amiga hardware FM, as mentioned last time. Nevertheless, some interesting textures can be produced.

In 'Both' mode, the combination can be heard as a mix if you have only mono audio, or as separate but simultaneous signals in your left (FM) and right (AM) stereo monitor. If you have mono only, make sure there is a 'Y' adaptor or some similar means to connect the output from both L & R Amiga audio sockets.

Programming with mouse control like this is a good way to explore the hardware characteristics, and possibly to perform live, but to be of greater use, the audio hardware would need to be controlled by a more elaborate program which itself incorporates some sort of sequence of sound events, or some other process beyond simple mouse-to-number mapping. This example should thus be considered a basis for more elaborate programs, where the data controlling the audio hardware might be generated by a composition algorithm, for example, or from screen maps like those in RGS. As with the Theremin example last time, there are some aspects of the code that may need further explanation, if it is to be used on this basis. The notes below are roughly in order of corresponding features in the code.

[My second program] demonstrates the modulation capabilities of the oscillators, both AM and FM. When you run the program, you can select AM, FM, or both, and then play notes with the mouse button, moving the mouse to control duration, frequency or volume.

I should point out that anything already explained in the Theremin example will not be repeated here. Also, there is insufficient space for complete details of the DMA hardware, and the rules constraining table length, sample period and output frequency; in fact, my examples do not obey all the rules, resulting in some loss in fidelity. If anyone pursuing this needs to reduce aliasing, or to have more accurate pitch control, please contact me individually.

Notes for Program Example 2:

1. The table length of various lookup tables is held as the parameter TabLen = 80 at the start of the listing. If you wish to experiment with other table lengths, then simply change this value from 80; everything else in the program is scaled to suit. The parameter ModCoeff = 100, although used once only, is likewise set at the start of the code, and can be easily fiddled for greater or lesser FM.

2. The carrier and modulator lookup tables are dimensioned as integer arrays, eg DIM CarWav% (TabLen / 2). However, it is only the modulator tables that hold 16-bit word values; the carrier table is compacted to 8-bit byte values as it is computed, but its length must still be expressed in word form, even to the DMA hardware which always fetches two bytes at once.

3. All four channels of audio DMA are used, so the symbol definitions for all the relevant latches and control

Program Example 2 - AmigaBasic audio modulation demo

```
REM Amiga audio AM and FM demo; mouse controls some parameters.
REM orig 30-Nov-89 Jim Sosnin
REM edit 25-Mar-90 JVS use arrays for audio hardware register addresses

TabLen = 80:            REM length of waveform lookup tables
ModCoeff = 100:         REM ModCoeff determines swing of carrier period.
MX = 0: MY = 0:         REM make space for vars used after VARPTRS renewed

DIM CarWav% ( TabLen/2 ), AModWav% ( TabLen ), FModWav% ( TabLen )
DIM AudLoc(4), AudLen(4), AudPer(4), AudVol(4), AudEn(4), FMod(4)

REM Carrier wavetable holds compacted byte values only, so 1/2 length.
REM AudLoc(4) etc arrays take place of earlier Aud0Loc, Aud1Loc, etc.
WINDOW 2, "AmigaBasic audio modulation demo"
GOSUB SymbolDef
GOSUB TableCalc
GOSUB MousePlay
WINDOW CLOSE 2
STOP: REM end of main program

REM ***********************************************
MousePlay:
PRINT "Amiga audio AM and FM demo. Press A or F, or B for both."
PRINT "Mouse vert. controls pitch or vol. (whichever is not being"
PRINT "modulated). Horiz. controls mod period. Press Q to quit."

WHILE 1
    WHILE MOUSE(0) = 0:      REM loop below until mouse button pressed
    KeyPress$ = INKEY$:      REM check if user has made another selection.

    IF KeyPress$ = "q" OR KeyPress$ = "Q" THEN
    PRINT "Quitting..."
    RETURN:                            REM escape 'WHILE' loops, return to main prog.
    END IF

    IF KeyPress$ = "a" OR KeyPress$ = "A" THEN
    PRINT "Amplitude Modulation"
    WhichChan = AudEn(0) + AudEn(1):  REM AM pair
    END IF

    IF KeyPress$ = "f" OR KeyPress$ = "F" THEN
    PRINT "Frequency Modulation"
    WhichChan = AudEn(2) + AudEn(3):  REM FM pair
    END IF

    IF KeyPress$ = "b" OR KeyPress$ = "B" THEN
    PRINT "Both AM and FM"
    WhichChan = AudEn(0) + AudEn(1) + AudEn(2) + AudEn(3)
    END IF
    WEND:                              REM lines below when mouse button pressed

    GOSUB DmaParams:                        REM renew VARPTRS just before
                                            REM each DMA enable
    POKEW ModCon, ( Set + AudEn(0) + FMod(2) ):  REM enable modulation
    POKEW DmaCon, ( Set + WhichChan ):             REM enable DMA

    WHILE MOUSE(0) < 0:           REM while mouse button pressed

    MX = MOUSE(1):                REM horiz and vert values
```

```
        MY = MOUSE(2)
        POKEW AudPer(0), MX * 40 + 124:    REM big number, longer AM env.
        POKEW AudPer(2), MX * 8 + 124:     REM FM mod. period shorter
        POKEW AudPer(1), MY + 124:         REM AM carrier period shortest
        POKEW AudVol(3), MY / 4:           REM FM carrier amplitude
        POKEW AudLen(0), 1:                REM for one-shot env (see text)
        WEND:                              REM lines below when mouse button released

        POKEW DmaCon, (Clr + WhichChan ):                    REM disable DMA
        POKEW ModCon, (Clr + AudEn(0) + FMod(2)):       REM disable modulation
WEND:                                      REM end of 'WHILE 1' loop
RETURN:                                    REM never gets here, returns on 'Q' above

REM  ***********************************************************

SymbolDef:

Abase = 14675968&:              REM Amiga audio hardware base address.
DmaCon = Abase + &H96:              REM Direct Memory Access control reg.
ModCon = Abase + &H9E:          REM Modulation Control reg. (AM & FM).
Set = &H8000:                       REM Set & Clr used with both above reg.
Clr = &H0

FOR i = 0 TO 3:                     REM set all 4 channel addresses & bitpats
        AudLoc(i) = Abase + &HA0 + &H10 * i:        REM location
        AudLen(i) = AudLoc(i) + 4:            REM length
        AudPer(i) = AudLoc(i) + 6:            REM period, in DMA ticks
        AudVol(i) = AudLoc(i) + 8:            REM volume
        AudEn(i) = 2 ^ i:               REM DmaCon DMA enable or ModCon AM enable
        FMod(i) = AudEn(i) * 16:              REM ModCon FM enable
NEXT i

WhichChan = AudEn(0) + AudEn(1):REM AM default to start
RETURN
REM  ***********************************************************

TableCalc:
PRINT: PRINT "Generating Wavetables .....";
AngInc = 3.14159 * 2 / TabLen:  REM angle increment for sine calc.

FOR i = 0 TO TabLen-1:
        SinVal = SIN ( AngInc * i ): SinInt% = 127 * SinVal
        FModWav%(i) = ModCoeff * ( 1 + SinVal ) + 124:      REM never < 124
        POKE VARPTR( CarWav%(0) )+i, PEEK ( VARPTR(SinInt%) + 1 )
        AModWav%(i) = 64 * ( 1 - i / TabLen ):                  REM falls from max 64
NEXT i

AModWav%(0) = 0:                REM 'cheat' for easy one-shot env.
PRINT "Wavetable generation complete.": PRINT
RETURN

REM  ***********************************************************

DmaParams:              REM mod tables hold 16-bit values, car tables 8-bit

        POKEL AudLoc(0), VARPTR ( AModWav%(0) )
        POKEW AudLen(0), TabLen
        POKEL AudLoc(1), VARPTR ( CarWav%(0) )
        POKEW AudLen(1), TabLen / 2
        POKEL AudLoc(2), VARPTR ( FModWav%(0) )
        POKEW AudLen(2), TabLen
        POKEL AudLoc(3), VARPTR ( CarWav%(0) ):             REM same table as for ch 1
        POKEW AudLen(3), TabLen 2
RETURN:                             REM             End of listing
```

registers would waste much paper if expressed individually. Instead, I have declared arrays, AudLoc(4) etc, for this purpose, keeping the names similar to the original Amiga definitions. An added advantage is that a sequencer algorithm, for example, could access all four channels in one loop. Look at the SymbolDef routine to see how the addresses and bitpats are computed.

4. The WINDOW 2, etc statement is more than a cosmetic enhancement over the default window which would otherwise greet the user; it is necessary to make the window active, so that AM, FM etc modes can be selected with the INKEY$ function.

5. You may notice the absence of GOSUB DmaParams from the main program. It is now called from within the MousePlay routine, just before DMA is enabled. I had too many problems with new variables displacing VARPTRs until I made this change; the remaining variables whose space needs to be pre-allocated are MX and MY, which are used after the mouse button is pressed.

6. The subroutine MousePlay has all the fun after SymbolDef and TableCalc have done their preparatory work. It sends a screen message to the user, then enters a 'WHILE 1' loop, from which the only escape is a 'Q' keypress. Within this outer loop there are four states, as in the Theremin program, but with the following additions:

The 'one-shot' method is perfectly suited for use in tune playing programs where it would be too much trouble to check if a particular note has finished, because a new note can be started at any time simply by reloading the correct AudLen() value, and any other parameters such as a new pitch.

(a) While the mouse button is up, the program loops through a check of the keyboard, in case the user has pressed a key to change mode or to quit. The variable WhichChan is set here, and used later as a parameter loaded into the DmaCon register; it is a compact way of representing any combination of the four AudEn(0 to 3) bit patterns.

(b) When the mouse button is pressed, DmaParams is called, as explained earlier, then the ModCon register has just two bits set, one to enable ch0 to modulate the amplitude of ch1, and the other to enable ch2 to modulate the period of ch3.

(c) In most programs, ModCon would need setting once only, at the start of the program, but here it is set, along with DmaCon, each time the mouse button is pressed; the reason is that both are reset whenever the button is released, not only to turn off the current sound, but to make sure that the DMA hardware is not left in an active state when the user chooses the 'Quit' option while the mouse is up.

(d) While the mouse button is held down, scaled mouse values are repeatedly poked into some DMA control registers, as in the Theremin code, and in addition, a length of 1 is poked into AudLen(0). This allows a 'one-shot' amplitude envelope to be generated with each mouse press, without having the program check to see when the note has finished, or activating interrupts. The DMA hardware reloads each channel's length counter from the corresponding AudLen() latch only when it gets to the end of a lookup table, or when DMA is first enabled; since Audlen(0) is poked with TabLen at each mouse press, in DmaParams, one complete envelope is generated, and on completion, when the DMA looks at AudLen(0) again, it finds 1 there. From that time until mouse release, the DMA hardware faithfully generates the AM signal as before, only now the amplitude is always zero, since I have 'wasted' the first AM table entry with the 'cheat' value zero. It is true that the DMA is busier than usual here, since the ch0 length counter is reloaded more often, but this hardware process is completely transparent to the user program, and has no speed penalty. The only inefficient code is the repeated poking of AudLen(0) with 1 in the mouse down loop; it really needs to be done once only, as far as the DMA is concerned, but I put it here so it would not wipe out the TabLen value prematurely.

(e) The 'one-shot' method above is used only for the AM pair; it is perfectly suited for use in tune playing programs where it would be too much trouble to check if a particular note has finished, because a new note can be started at any time simply by reloading the correct AudLen() value, and any other parameters such as a new pitch. The alternative involves enabling interrupts, or at least polling an interrupt request register, and will not be discussed here. However, if you wish to articulate the note only by enabling or disabling DMA (pressing or releasing the mouse button, in this program), then remove or comment out this one-shot line, discussed above.

7. In the SymbolDef routine, addresses are set in a loop to save space, as already mentioned, with most symbol names retaining some similarity with original Amiga names. The exception is the ModCon register, for which the Amiga name is ADKCON, but I had already changed this to ModCon in the Theremin example. Note the method used to set AM as the start default.

8. In the TableCalc routine, the carrier table compacts 16-bit words to 8-bit byte entries, as before, but both AM and FM modulator tables hold full 16-bit values. These values must stay within the same ranges that apply to program-generated numbers for controlling volume or period: volume is in the range 0 to 64 (yes, 65 values), and period must be no less than 124. In fact, the period should range between 124 and 256 for best fidelity, which requires longer lookup tables for lower octaves; I have ignored this rule here. The FM table does not generate 'true' FM, because its entries control the period, rather than its inverse, the sampling rate (or increment) usually used for FM. The

resulting nonlinearity is perceived as a change in centre frequency of the carrier, especially noticeable if an enveloped modulator is used. It may be possible to cancel this with a compensating inverse function while writing the table, but I have not explored this yet.

9. The same carrier table is able to be used for both AM and FM because the DMA hardware uses separate table pointers for the four channels, so they don't care which block of memory they are stepping through. If you wish to use a harmonically richer signal for the AM pair, it is a simple matter to dimension a new array name, and fill each element with a sum of sine values; just change the SinInt% equation, scaling it so the sum does not exceed ±127. If you wish to change the amplitude envelope, you will probably have to do it in a separate loop, rather than squeezing it into the common one as I have; values must range from 0 to 64. The tables can all have different lengths, too.

10. Some final notes about modulation restrictions:

(a) It is possible to have any combination of the 8 LSBits set in the ModCon register, but bits 3 and 7 are redundant, as ch3 can only function as a carrier; ch0, 1 & 2 can each function as a carrier or can modulate the next higher channel. If both AM and FM control bits are set for a particular modulator/carrier combination, then alternate words of the modulator table are used to modulate the amplitude, then period, of the carrier; my example does not do this, however, but has independent AM and FM pairs.

(b) If a carrier has either its volume or period modulated, then that parameter cannot simultaneously be controlled by the program, but the unmodulated parameter can.

(c) A modulator cannot have its 'volume', ie the amount of modulation it produces, controlled by the program; its lookup table values are always transferred directly by the DMA. Its period can be controlled, either by the program as in this example, or by another modulator, although I see no immediate musical use for this, given the FM limitations already mentioned.

Despite these restrictions, it is still possible to produce sounds with dynamic spectra in real time because the DMA takes most of the time-critical load off the user program. If polyphony is not required, the user program can start a note, then control the DMA to crossfade between four channels, each of which is scanning a table with same period and length, but different content. For polyphony, the user program could start independent notes on all four channels, or on 2 AM pairs, then control the DMA to adjust the starting positions within much longer tables, which have had many cycles of varying waveforms written into them. I will present examples of these techniques in a future issue.

References

Amiga Hardware Reference Manual, Addison-Wesley Inc.
Amiga Basic (Microsoft Basic for the Amiga), Commodore-Amiga Inc.

Australian Computer Music Association, Inc.
1990 Membership Application

The annual membership fee is $10 (individuals, $100 institutions/organisations). Please complete the details below and forward to:

The Secretary
Australian Computer Music Association, Inc.
P.O. Box 4136
Melbourne University 3052

Cheques should be made out to the Australian Computer Music Association, Inc. Membership entitles you to receive and contribute to Chroma, the Association's quarterly newsletter and participation in all ACMA sponsored events.

Name:

Address:

Telephone: (   )

Organization:

Particular areas of interest / equipment / software used:

What computers do you use?

Do you agree to allow your name, telephone number and interests to be circulated to other members of the Association?

Yes ☐ No ☐
Tick here if this is a Renewal ☐

Signature: Date:

Back Issues of Chroma are available at $2 each (Double issues $4).

Member No:
Receipt Date: